THE BLOCK ADAPTIVE MULTIGRID METHOD APPLIED TO THE SOLUTION OF THE EULER EQUATIONS[1]

Nikos Pantelelis
PhD Student, National Technical University of Athens
P.O.Box 64078, 15710 ATHENS, GREECE

SUMMARY

In the present study, a scheme capable of solving very fast and robust complex nonlinear systems of equations is presented. The Block Adaptive Multigrid (BAM) solution method offers multigrid acceleration and adaptive grid refinement based on the prediction of the solution error. The proposed solution method was used with an implicit upwind Euler solver for the solution of complex transonic flows around airfoils. Very fast results were obtained (18-fold acceleration of the solution) using one fourth of the volumes of a global grid with the same solution accuracy for two test cases.

INTRODUCTION

Although multigrid methods were introduced as grid adaptation techniques they have been established only as fast and efficient solvers for large scale computational problems. Up until today only a few adaptive multigrid schemes have been presented, e.g. the Multilevel Adaptive Technique MLAT (ref. 1), the Fast Adaptive Composite grid method FAC (ref. 2) and others (ref. 3), but the domain of applications has been mainly restricted to the solution of elliptic type equations. Regarding the development of adaptive schemes for hyperbolic systems of equations, only a few attempts have been made to take advantage of the favourable multigrid concept for the acceleration of the solution. On the other hand great advantages have been pointed out for the use of the truncation error prediction as a reliable error sensor for grid adaptation, though few studies exhibited numerical proofs (ref. 4,5).

The present study, a dynamically grid adaptive method, namely the Block Adaptive Multigrid (BAM) method, is presented, incorporating a reliable device for the prediction of the error and a composite multigrid solver. The method is based on a Full Multigrid scheme: starting from an acceptable coarse mesh the solution creates finer grid patches with block grid refinement. The refined regions are grouped into rectangular blocks defining a composite structure which is totally handled by the multigrid method. In this way an adapted non uniform domain is decomposed into regular subdomains solved with common solvers. Further, the communication between blocks and the

[1]This work was partially supported by CEC/ Brite-Euram contract AERO-0018C.

parallelization of the BAM method are based on domain decomposition ideas. For the integration and relaxation of the time marching Euler equations, an unfactored implicit upwind finite volume scheme is employed. The proposed method is tested for two complicated transonic inviscid cases for two different airfoils. Using the proposed method stable and accurate results are obtained in a small number of work units (18-fold acceleration with 4.5 times fewer volumes).

FINITE VOLUME DISCRETIZATION

The general inviscid flow is described by the Euler equations that can be solved using the very popular time-marching conservative formulation. For the two dimensional case, conservation laws are used with body- fitted coordinates ξ, η:

$$\frac{\partial U}{\partial t} + \frac{\partial E}{\partial \xi} + \frac{\partial F}{\partial \eta} = 0 \qquad (2.1)$$

The steady state solution is found when the time derivative of the solution vector becomes negligible. The solution vector and the fluxes normal to ξ=const. and η=const. faces are given respectively:

$$U = J \cdot \bar{U} \quad \text{and} \quad E = J \cdot (\bar{E}\,\xi_x + \bar{F}\,\xi_y)\,, \quad F = J \cdot (\bar{E}\,\eta_x + \bar{F}\eta_y) \qquad (2.2)$$

At the Cartesian coordinate system the corresponding solution vector and the inviscid fluxes are given by:

$$\bar{U} = \begin{bmatrix} \varrho \\ \varrho u \\ \varrho v \\ e \end{bmatrix} \quad \text{and} \quad \bar{E} = \begin{bmatrix} \varrho u \\ \varrho u^2 + p \\ \varrho u v \\ (e+p)u \end{bmatrix}, \quad \bar{F} = \begin{bmatrix} \varrho v \\ \varrho u v \\ \varrho v^2 + p \\ (e+p)v \end{bmatrix} \qquad (2.3)$$

In equation (2.3) e is the total energy ($e = \frac{p}{(\gamma-1)} + \frac{1}{2}\varrho[u^2+v^2]$), p and ϱ are the pressure and the density respectively and J is the Jacobian of the inverse mapping.

For the discretisation of equation (2.1) a cell-centered finite volume method is used. For the pseudo-time evolution a Newton linearization scheme is adopted which, being an implicit scheme, allows high CFL numbers (100-200) to be used with local time stepping (different Δt for each volume):

$$\frac{\Delta U}{\Delta t} + (A^n\,\Delta U)_\xi + (B^n\,\Delta U)_\eta = -(E^n_\xi + F^n_n) \qquad (2.4)$$

Or else:

$$L^n \cdot \Delta U = \left(\Delta t^{-1} + A^n_\xi + B^n_\eta\right) \cdot \Delta U = -\,\mathrm{Res}^n(U^n) \qquad (2.5)$$

Where ΔU is the correction of the solution vector, A and B are the Jacobians

of the fluxes E and F respectively for the time level n:

$$U^{n+1} = U^n + \Delta U \quad , \quad A^n = \left(\frac{\partial E}{\partial U}\right)^n \text{ and } B^n = \left(\frac{\partial F}{\partial U}\right)^n \tag{2.6}$$

Upwind differencing of the flux vectors is used to achieve a diagonal dominant system of equations. Thus, using a symmetric collective point Gauss-Seidel relaxation scheme very good smoothing properties are attained for the multigrid calculations. For the flux calculations a linear locally one-dimensional Riemann solver (Godunov- type) is employed thus, the homogeneous property of the Euler fluxes [7] is guaranteed. The mean values of the conservative variables at both sides of the faces are used as input variables for the Riemann solver. For the calculation of the fluxes E and F the conservative variables are extrapolated using an upwind characteristic variable interpolation method (MUSCL- type). The interpolation scheme uses two volumes from both sides of each face. The accuracy of the scheme raises up to third order depending on the sign of the local eigenvalues of the Jacobians A and B. The local accuracy of the finite volume method is sensor- controlled so the monotonic behaviour of the solution is guaranteed. Boundary conditions are required for both sides of eq.(2.4), so for the RHS of eq.(2.4) characteristic boundary conditions are extracted from the Riemann solutions at the wall and at free surfaces. For the LHS of eq.(2.4) simple boundary conditions are prescribed on the ΔU in phantom shells.

THE BLOCK ADAPTIVE MULTIGRID METHOD

The Block Adaptive Multigrid (BAM) method is composed of three main parts: the fast nonlinear multigrid solver, the truncation error approximation for the prediction of the solution error and the block composite grid solver. Additionally an efficient solution strategy is required in order to achieve fast and robust accurate solutions.

Multigrid Implementation

Concerning the multigrid method the Full Approximation Scheme (FAS) is employed as it is better than the Correction Scheme (CS) for Newton linearisation schemes. FAS has the major advantage in that it operates with the same solution vector of the initial algorithm so it is best suited for the solution of composite grid structures. The efficiency and the performance of the multigrid implementation are maintained adopting the "alternate point of view" of the FAS (ref. 1). Following this approach the finer grid levels are considered as devices to increase the spatial accuracy of the solution whereas the coarser grid levels are devices to accelerate the solution. The formulation of the solution is independent of the grid level (coarse or fine) and the type of grid (local or global) by simply adding to the RHS of eq.(2.5) the appropriate fine-to-coarse defect correction (τ). For the enumeration of the multigrid levels the classical mode is adopted. Thus, the current grid

level is denoted by n, its next finer level by n+1 and its corresponding finest grid level by m ($m \geq n \geq 1$) (1 is the coarsest grid level of the domain). The solution formulation for the current grid level n is given as:

$$L_n \cdot \Delta U_n = - Res_n(U_n) + \tau_n^{n+1} \tag{3.1}$$

considering that the fine-to-coarse defect correction is:

$$\tau_n^{n+1} = \Sigma_n^{n+1}\left[L_{n+1} \cdot \Delta U_{n+1} \right] - L_n \cdot (I_n^{n+1} \Delta U_{n+1}) \quad \text{and} \quad \tau_m^{m+1} = 0 \tag{3.2}$$

Because one multigrid cycle equals one time step the time scale is not considered.

Because of the applied cell centered finite volume scheme the cellwise coarsening is adopted; each coarse volume is constructed by four consequently finer ones. The cellwise coarsening maintains the outer edges of the volumes (straight implementation of conservation laws) but the coarse grid centers are not a subset of the fine ones. Therefore, two different restriction operators are required. The restriction operator (I) for the physical variables is the simple average over the four fine volumes:

$$\Delta U_n = I_n^{n+1} \Delta U_{n+1} = \frac{\sum \Delta U_{n+1}}{4} \tag{3.3}$$

In contrast, the restriction operator (Σ) for the generalized residuals, Res and τ, is the summation of the residuals of the corresponding fine volumes. The fluxes of the inner common fine grid faces are canceled, thus flux conservation at the coarser grid levels is maintained:

$$Res_n = \Sigma_n^{n+1} Res_{n+1} = \sum Res_{n+1} \tag{3.4}$$

For the reverse direction of the multigrid cycle (coarse-to- fine direction) neither Euler solutions nor relaxation sweeps are required. Therefore, ΔU variables are stored for all grid levels and only these are prolongated from the coarse-to-fine direction using the standard FAS prolongation formulation:

$$\Delta U_{n+1} = \Delta U_{n+1} + II_{n+1}^{n} (\Delta U_n - I_n^{n+1} \Delta U_{n+1}) \tag{3.5}$$

For the prolongation operator (II) simple injection is adopted, i.e.:

$$U_{n+1} = II_{n+1}^{n} U_n = U_n \tag{3.6}$$

Due to the composite grid structure, complicated interpolation schemes would have increased considerably the programming complexity with minor advantages. For the present multigrid implementation the V-cycle is applied with the improvement of increasing the number of relaxation sweeps as coarser levels are processed.

Solution Error Approximation

To determine the erroneous regions of the computational domain where

increase of the accuracy is required, a reliable error indicator is required. Two approaches essentially exist: the first one is the physically based information of the problem, i.e. solution gradients, while the second one, numerically motivated, is the evaluation of the discretization error. The former approach may implicate the refinement process to reduce errors that have no influence on the global solution. In contrast, the evaluation of the truncation error approach indicates errors which can be confronted by the refinement procedure. On the basis of the Richardson extrapolation an estimation of the truncation error is formulated as:

$$T_n = Q(h) \cdot u \tag{3.6}$$

Whereas for a uniform Cartesian mesh holds:

$$T_n = c \circ (h)^p \tag{3.7}$$

In equations (3.6), (3.7) Q is the differential operator, u is the physically correct solution, h is the mesh size of the finest grid level, p is the local order of accuracy and c is an unknown grid independent factor. Guided by the physical interpretation of the truncation error and the Richardson extrapolation concept, the difference of residuals between the two finest grid levels is used for the truncation error calculation. This error sensor provides a reliable local estimation of the solution error. Although the fine-to-coarse defect correction of eq.(3.2) is directly provided by the multigrid solution it was proved to be an unreliable error indicator. Fortunately the use of the Res(U) operator instead of the $L \circ \Delta U$ operator gives very good results. The explanation is that due to the Newton linearization scheme the Res(U) differential operator is insensitive to relaxation errors maintaining the accuracy of the solution. Therefore, the solution error evaluation for the grid level m-1 (m is the current local finest grid level), is given by:

$$T^{m}_{m-1} = \Sigma^{m}_{m-1} T_m - T_{m-1} = \Sigma^{m}_{m-1} Res_m(U_m) - Res_{m-1}(I^{m}_{m-1} U_m) \tag{3.8}$$

For a totally converged solution equation (3.8) reduces to:

$$T^{m}_{m-1} = - Res_{m-1}(I^{m}_{m-1} U_m) \tag{3.9}$$

As this truncation error sensor is a vector, a reduction norm to a single value is required. At the present study the Euclidean norm has been adopted because it shows similar distribution to the pressure error of the solution. The proposed error sensor requires additional work of only one fourth of a simple flux calculation and it does not demand totally converged solution ($Res_m(U_m) \neq 0$) although it converges from the early time steps to the steady state.

The prediction of the solution error for the new finer grid level m-1 can be computed only if the assumption of a uniform Cartesian mesh is adopted ($h_{m+1} = h_m/2$). Thus following equation (3.7) we get:

$$T^{m+1}_{m} = 2^{-p} \; T^{m}_{m-1} \tag{3.10}$$

For the determination of the order of accuracy (p) of eq.(3.10), which varies from one to two depending on the flow features, the sensing functions that are used to control monotonicity at the integration routine are also used to calculate the local accuracy of the scheme. Unfortunately, equation (3.10) holds only for Cartesian grids, thus using this equation in arbitrary grids

errors are expected to the predicted error levels.

Composite Grid Structure and Solution Procedure

For the optimal approximation of the most accurate solution within the minimum amount of work, several grid adaptive techniques and structures have been developed. The composite grid structure has many advantages by enabling the solution of a globally non uniform grid using a union of locally uniform subgrids (blocks) (ref. 2). Subgrid uniformity is essential to assure multigrid efficiency using simple solution routines (similar to the single grid solver). Moreover, significant simplifications to the data structure and to the interface manipulation are attained when the blocks are restricted to rectangles in the computational domain and the grid refinement ratio is 2 for each refined direction (ref. 5).

In a composite multigrid method the major problem is the requirement of special manipulation at the artificial boundaries to maintain the accuracy of the global solution. To suppress errors inconsistent with the solution method at the artificial boundaries (two fine volumes share the same edge with a coarse one) certain special boundary conditions must hold. Namely: accuracy preservation of the integration scheme, flux conservation and flux- splitting compatibility with respect to the global grid solver. To preserve the accuracy of the solution across an artificial boundary, the fluxes of the domain should be calculated at the block's finest grid level. For these calculations the standard integration routine has been employed. The basic idea is to properly construct false fine volumes at the coarse grid boundaries using all the available information near the interfaces. The modified scheme at the intergrid boundaries is depicted in fig.1 and fig.2 for the one and two dimensional analogs, respectively. Following these analogs the flux calculations should be started from Block 1 including the boundary interfaces. For the evaluation of the virtual volumes F3 and F4 (fig.1) the coarse volumes C1, C2 and the fine ones F1, F2 are considered adopting the same MUSCL- type extrapolation scheme used by the integration routine. At the two dimensional analog, the same couple of coarse volumes is used with the corresponding fine volumes (fig.2). In this way, the order of accuracy of the global solution scheme is maintained. After calculating the fluxes of the finest subgrid (Block 1 in fig.1), the boundary fluxes of the neighboring coarser subgrids (Block 2) can be calculated explicitly using conservation of fluxes. According to the multigrid restriction operator for the residuals, flux conservation across the artificial boundaries is achieved by addressing the summation of the fluxes of the fine volume faces to their adjacent coarse volume face (fig. 2).

With respect to the relaxation procedure, the modifications to the flux vector splitting and the relaxation scheme at the subgrid interfaces are totally handled by the multigrid algorithm. Adopting the "horizontal" communication mode among subgrids, the fine subgrids (i.e. block 1) are relaxed at the first grid level (fig.1). At the next multigrid level the coarse subgrids (i.e. block 2) are relaxed together with the restricted fine ones, while the block structure of the composite grid is preserved throughout the multigrid cycle.

Due to the block structure, the composite grid has the advantage of a straightforward implementation of vectorization and parallelization. This can be achieved when subgrids are considered totally independent from each other introducing ideas from the domain decomposition theory. Concerning the "vertical" communication mode (ref. 11), each subgrid is solved and relaxed independently for the complete multigrid cycle whereas data exchange among subgrids is permitted only at the start and/or the end of each cycle. The computational domain decomposition (different from the composite block structure) (ref. 8) should be performed according to load balancing and vectorization criteria taking into consideration the hardware configuration. The "vertical" communication mode is preferred from the "horizontal" mode when we deal with parallel processing and is used even though the latter mode has better convergence rates. Using the "vertical" mode the idle and communication time among processors can be considerably decreased.

With respect to the dynamically adaptive multigrid strategy a modified Full Multigrid scheme is applied, starting with a global coarse grid of acceptable grid resolution. After convergence (or after a fixed amount of work) at the current grid, the truncation error is calculated and the solution error is predicted. In regions where the prediction of the error is above a threshold the corresponding volumes are flagged and grouped into rectangular blocks. Afterwards, the domain is decomposed to the appropriate blocks where only those which contain the flagged volumes are refined to the next grid level injecting the coarse grid solution to the refined grid. The refinement procedure continues until the entire computational domain has local truncation errors below a given threshold. Clearly this strategy has the benefit of a continuous iterative procedure without wasting CPU-time on calculations that will not be used after the mesh refinement. Taking advantage of the most accurate available solution the proposed scheme converges straight to the most efficient solution.

RESULTS AND DISCUSSION

In order to verify the accuracy and to validate the efficiency of the proposed method two transonic inviscid test cases were investigated. The first case is a NACA- 0012 airfoil for Mach 0.80 at angle 1.25 degrees while the second case is a RAE- 2822 airfoil for 0.73 Mach at 2.79 degrees. A Work Unit (W.U.) is defined by the CPU-time required for a global finest grid relaxation sweep in lexicographic order while for a single grid running one time step costs four work units.

For both test cases the grid adaptation procedure as described above was applied. Starting point is two global multigrid levels with 64x14 volumes at the finest grid level. The user supplies only the maximum number of the additional grid levels which for both test cases were the same: two refinement grid levels. The truncation error threshold was defined explicitly targeting to a three- fold reduction of the initial error levels. For the convergence criterion the Euclidean norm of the correction vector was employed.

For the first test case (NACA-0012) a comparison between the computed and the predicted truncation error contours is given in figures 3 and 4, resp.. Some differences, not crucial, are due to the incorrect evaluation of the

local accuracy of the solution (p). For the same grid level the actual pressure error contours and the entropy contours in figures 5 and 6 are included. A comparison between the truncation errors and the pressure errors shows their similar distributions. Following the proposed method very accurate results were obtained in comparison to the global solutions as depicted in fig.7. In fig.7 the Mach- distributions along the airfoil are depicted for two global grids (256x56 and 128x28) and one composite grid sharing both grid levels. A comparison of these Mach distributions shows that at regions with the same mesh size the solutions between a global and a local refinement coincide. Additionally, in fig.8 the Mach contours together with the composite grid are given. In fig.9 the efficiency of the BAM method with respect to the single grid and the global multigrid schemes is clearly indicated. A 19-fold and 4-fold acceleration were achieved with respect to the single grid and to the multigrid cases, respectively. The final grid adaptive solution requires 4.5 times fewer volumes (from 14336 to 3200) for practically the same accuracy with a globally refined grid (0.35% discrepancy of the computed Cl).

Similar efficiency was achieved for the second test case. The domain decomposition into subgrids took place after 50 time steps spent on coarser grid levels. Using the same truncation error threshold as in the previous test case, a 17-fold acceleration was achieved with respect to the single grid and a 4.7- fold reduction of the volumes (from 14336 to 3056) for practically the same accuracy (Cl discrepancy is 0.2 %). The convergence histories of the error reduction and the lift coefficient are shown in fig.10 while in fig.11 the final composite grid together with the isomach contours are depicted. It is important that throughout the solution process the multigrid convergence rates were maintained while the overhead for the interface computations was negligible, i.e. only 2 % for a nine block structure with respect to an equivalent global grid.

As no parallel machine was currently available a simulation of the parallelization for the procedure has been attempted in order to evaluate the performance of the two different communication modes for the BAM method. To achieve this, the data exchange was properly adjusted among subgrids according to each communication mode. For the "horizontal" mode (semi-parallel) and the "vertical" mode (parallel) the convergence histories are shown in fig.12. A comparison between the parallel modes and the sequential BAM method shows only a small reduction of the convergence rates for the parallel ones. What is further required for a parallel implementation is a computational domain decomposition of the composite structure based on specific load balancing criteria in order to reduce the idle time among processors.

CONCLUSIONS

The great advantages of the Block Adaptive Multigrid (BAM) method were exhibited. The incorporation of numerous efficient schemes into the BAM method makes the ultimate target of solving complex problems in just a few work units feasible. At the same time robustness, simplicity and accuracy of the single grid code were maintained at the new method. The extension to viscous and hypersonic three dimensional problems is straight forward though semi coarsening multigrid can be also included. On the other hand, in order to improve the adaptation capabilities, a moving grid point scheme should also be

considered as the grid alignment towards certain flow features is essential in some problems in combination with the present grid refinement procedure.

Although the basic features of the BAM method have been determined and verified, a few other issues remain to be settled. The first is the construction of a data structure which will handle more efficiently the block structure of the composite grid. The second is to approximate more precisely the local order of accuracy of the solution in such a way that the prediction of the solution error would be more accurate. Additionally, the implementation of the BAM method to other solution algorithms and equations is foreseen as the BAM method was designed in the general concept of the finite volume method.

REFERENCES

1. A.Brandt: Multi-level Adaptive solutions to boundary-value problems, Math. Comp., 31, pp. 333-390.

2. S.McCormick: Multilevel Adaptive Methods for Partial Differential Equations (SIAM, Philadelphia, 1989).

3. W.Joppich: A multigrid algorithm with time-dependent, locally refined grids for solving the nonlinear diffusion equation on a nonrectangular geometry, in Proceedings of the 3rd European Conference in Multigrid methods, eds W.Hackbusch and U.Trottenberg, Int.Ser.Num. Math, 98 (Birkhaeuser Verlag, Basel, 1991) pp. 241-252.

4. K.Y.Fung, J.Tripp and B.Goble: Adaptive refinement with truncation error injection, Comp.Meth.Appl.Mech.Engng., 66 (1987), pp. 1-16.

5. M.J.Berger and A.Jameson: Automatic adaptive grid Refinement for the Euler Equations, AIAA J., 23 (1985), pp. 561.

6. A.Eberle: Characteristic flux averaging approach to the solution of the Euler equations, VKI Lecture series 1987.

7. A.Kanarachos and N.Pantelelis: Block Full Multigrid Adaptive Scheme for the Compressible Euler Equations, Proceedings of the 13th Intl.Conf. on Numerical Methods in Fluid Dynamics,eds. M.Napolitano, F.Sabeta, Lecture Notes in Physics V. 424 (Springer Verlag, Berlin, 1993) pp.265-269.

8. J.Scroggs and J.Saltz: Distributed Computing and Adaptive Solution to Nonlinear PDEs, Fourth International Symposium on Domain Decomposition Methods for Partial Differential Equations, eds. R. Glowinski et al, (SIAM, Philadelphia, 1990) pp.409-417.

9. C.Liu: The Finite Volume Element (FVE) and Fast Adaptive Composite Grid Methods (FAC) for the incompressible Navier-Stokes Equations, Proceedings of the Fourth Copper Mountain Conference on Multigrid Methods, eds. J.Mandel et al. (SIAM, Philadelphia, 1989) pp.319-337.

10. M.A.Heroux and J.W.Thomas: TDFAC: A composite Grid Method for Elliptic Equations, Proceedings of the Fourth Copper Mountain Conference on Multigrid Methods, eds. J.Mandel et al. (SIAM, Philadelphia, 1989) pp.273-285.

11. Y.Yadlin and D.A.Caughey: Block Multigrid Implicit Solution of the Euler Equations of Compressible Fluid Flow, **AIAA J.**, Vol.29 (1991), pp. 712-719.

12. M.Lemke and D.Quinlan: Local Refinement Based Fast Adaptive Composite Grid Methods on SUPRENUM, Multigrid Methods: Special Topics and Applications II, GMD- Studien Nr. 189, eds. W.Hackbusch and U.Trottenberg (GMD, Sankt Augustin, 1991), pp. 179-189.

13. M.Giles: Accuracy of Node-Based Solutions on Irregular Meshes Proceedings of the 11th Intl.Conf. on Numerical Methods in Fluid Dynamics,eds. D.Dwoyer et al., Lecure Notes in Physics V.323 (Springer Verlag, Berlin, 1989) pp.273-277.

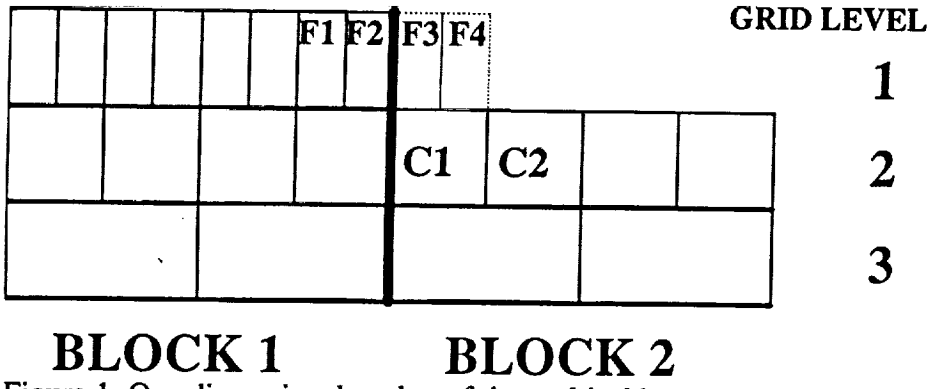

Figure 1. One dimensional analog of the multigrid coarsening and the fictitious volumes F3.F4 for the intergrid flux calcultations.

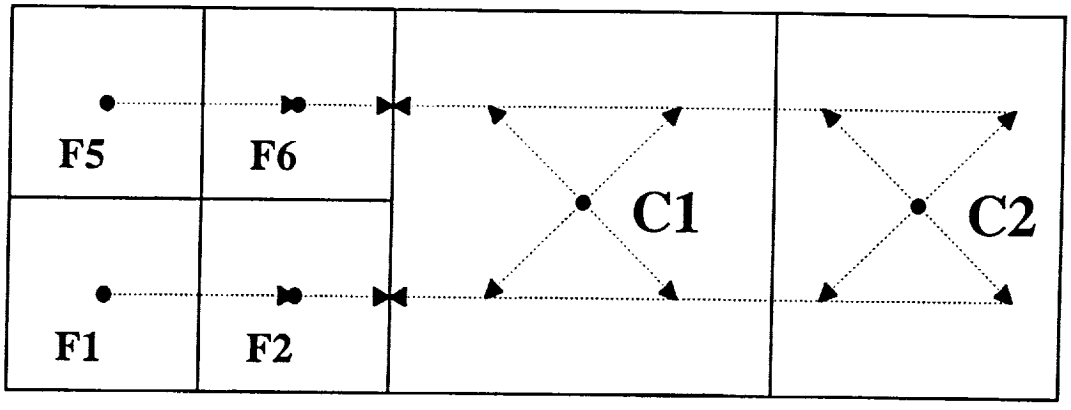

Figure 2. Two dimensional analog of the special variable interpolation procedure at the integrid boundaries.

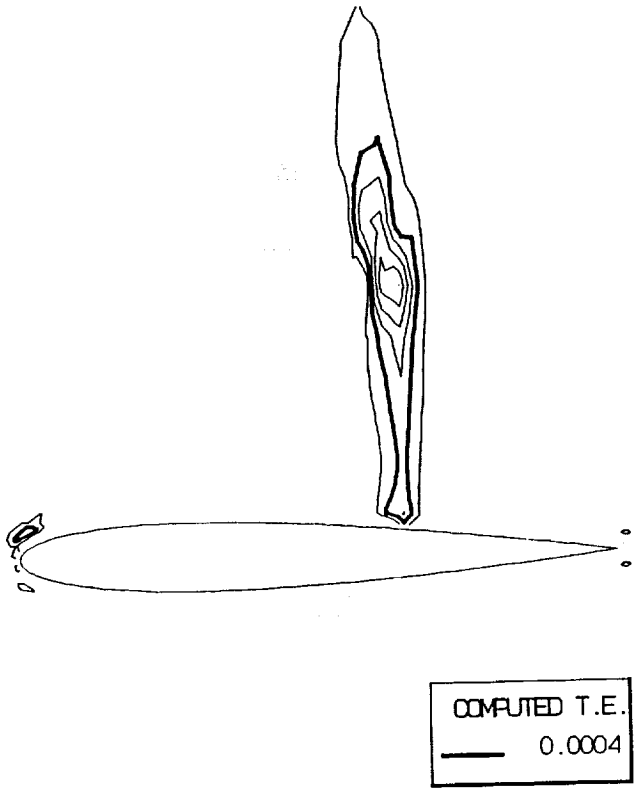

Figure 3. Computed truncation error contours for the finer grid level, CASE 1.

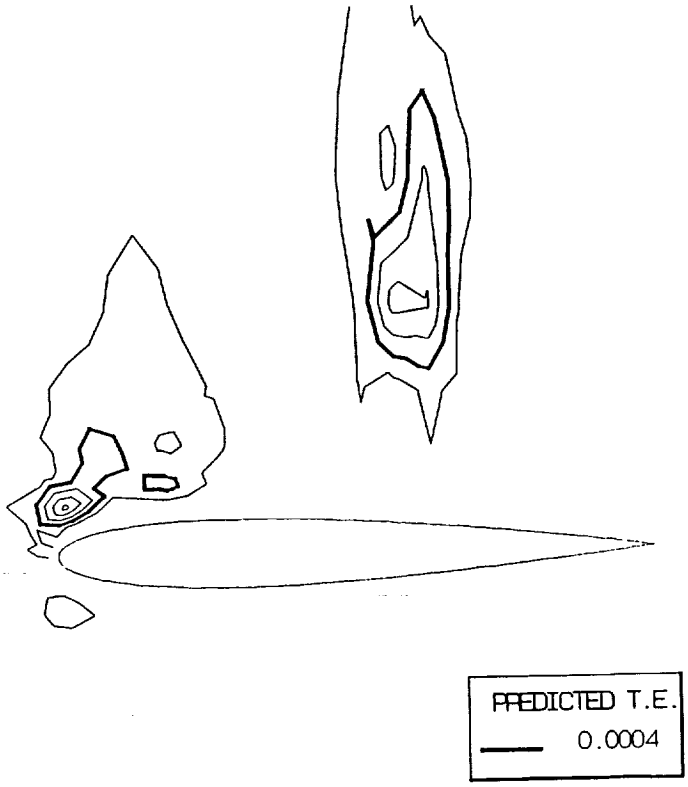

Figure 4. Predicted truncation error contours for the finer grid level, CASE 1.

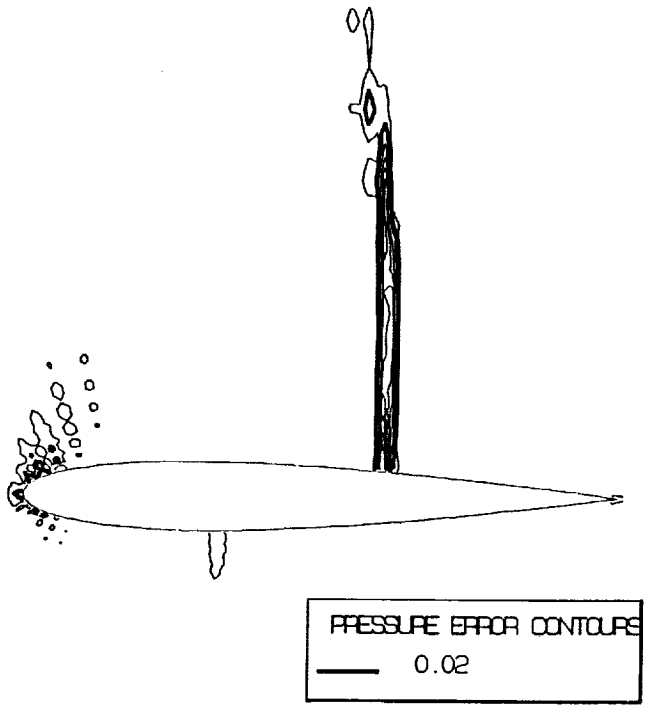

Figure 5. Pressure error contours for the finer grid level, CASE 1.

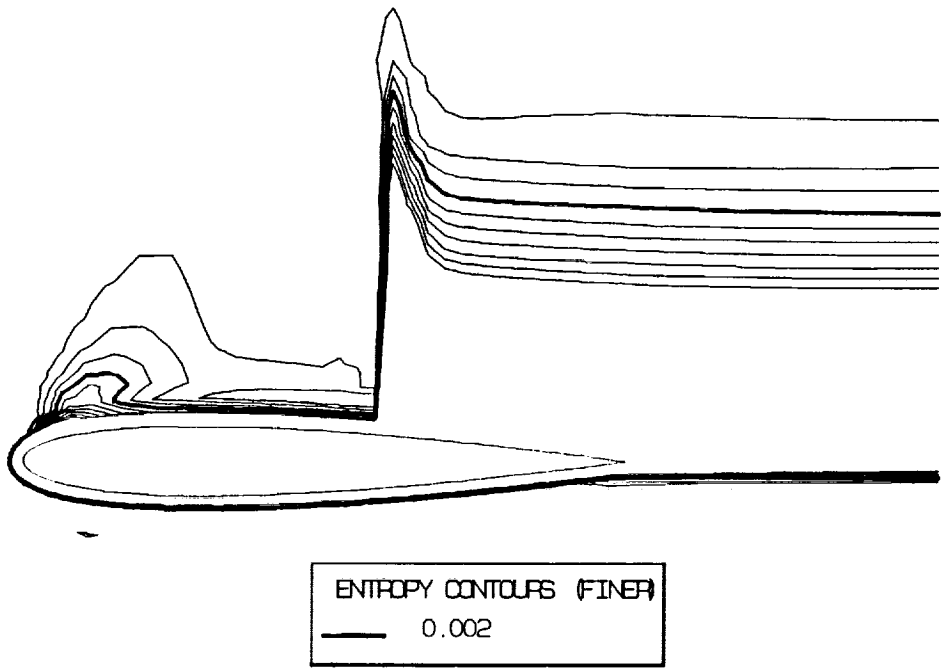

Figure 6. Entropy contours for the finer grid level, CASE 1.

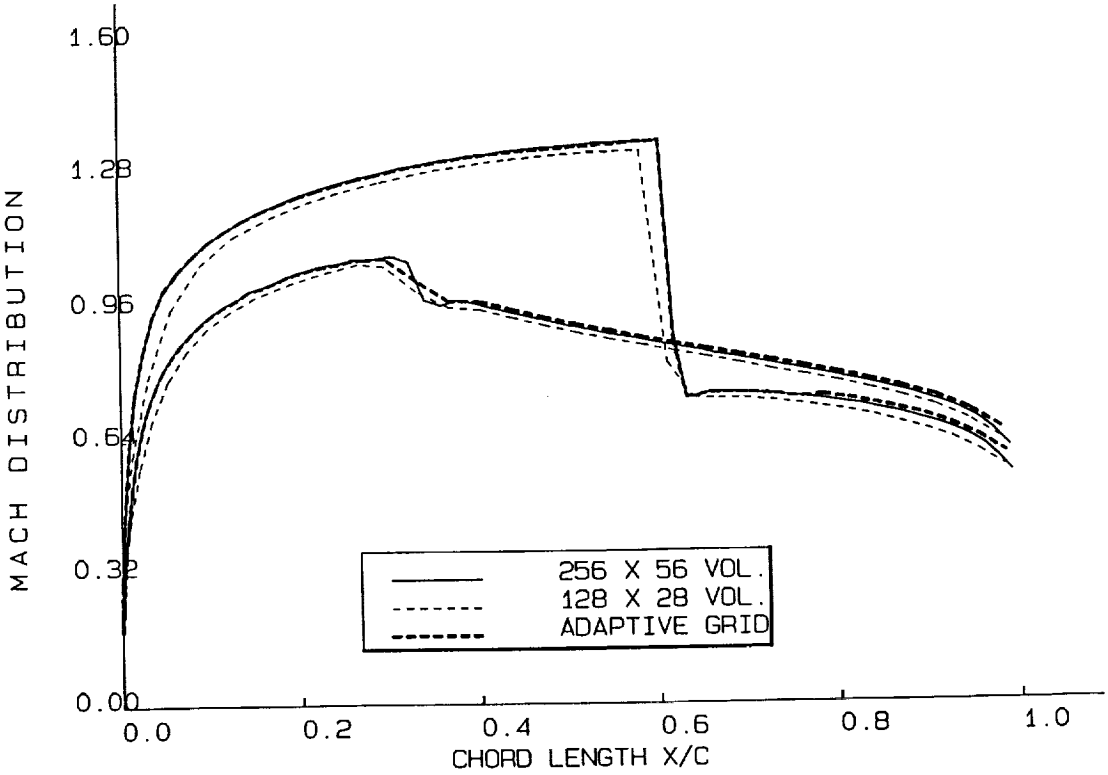

Figure 7. Mach distribution along the airfoil for a global fine, a global coarser and an adaptive composite grid (CASE 1).

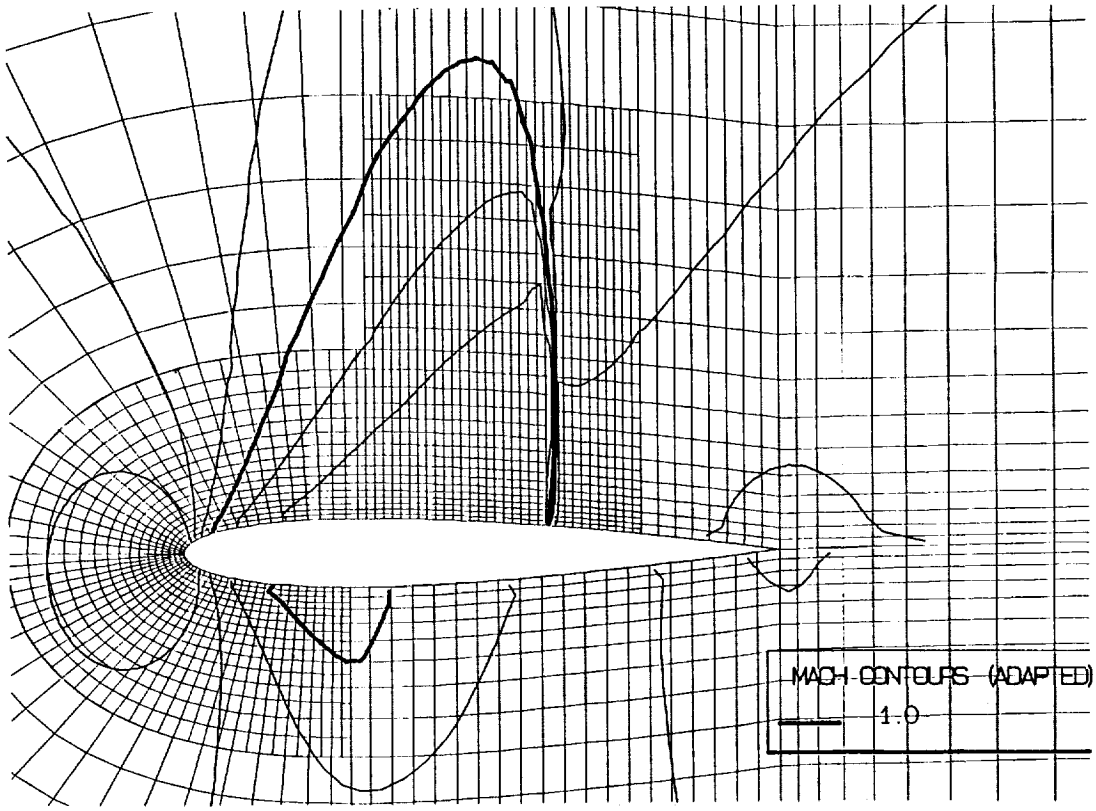

Figure 8. The adaptive composite grid and the corresponding Mach contours (CASE 1).

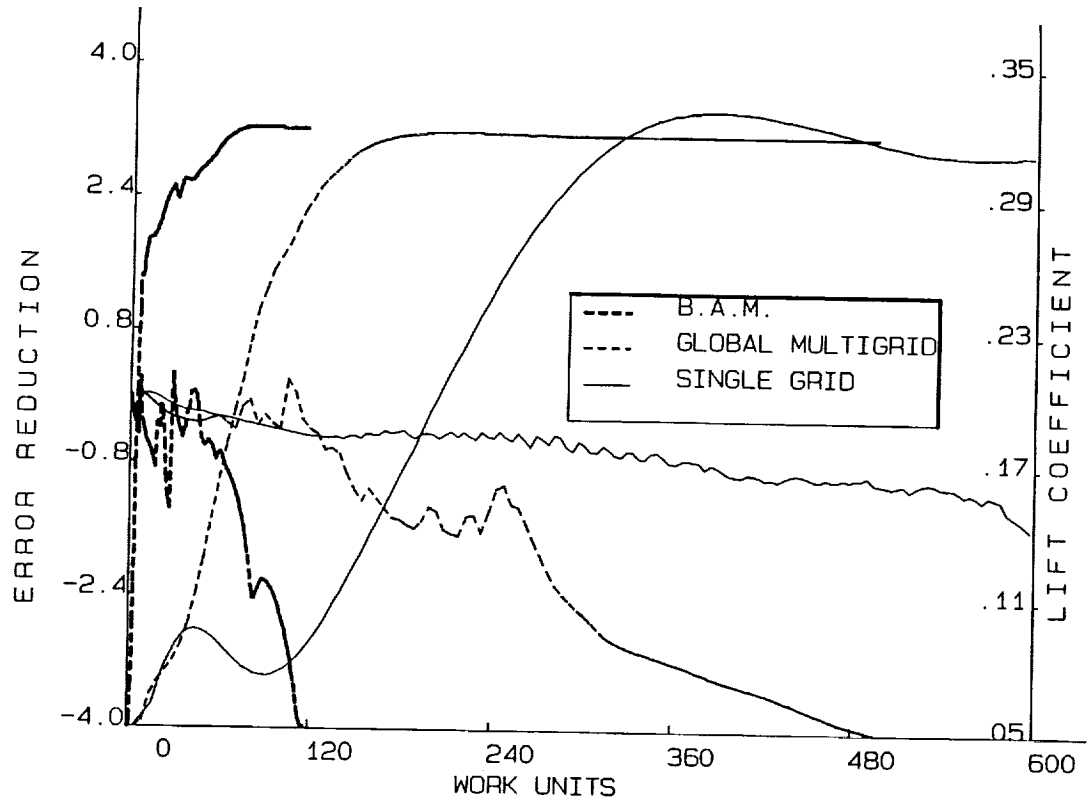

Figure 9. Convergence histories of the lift coefficient and the logarithmic Euclidean error reduction for the single grid, global multigrid and B.A.M. method (CASE 1).

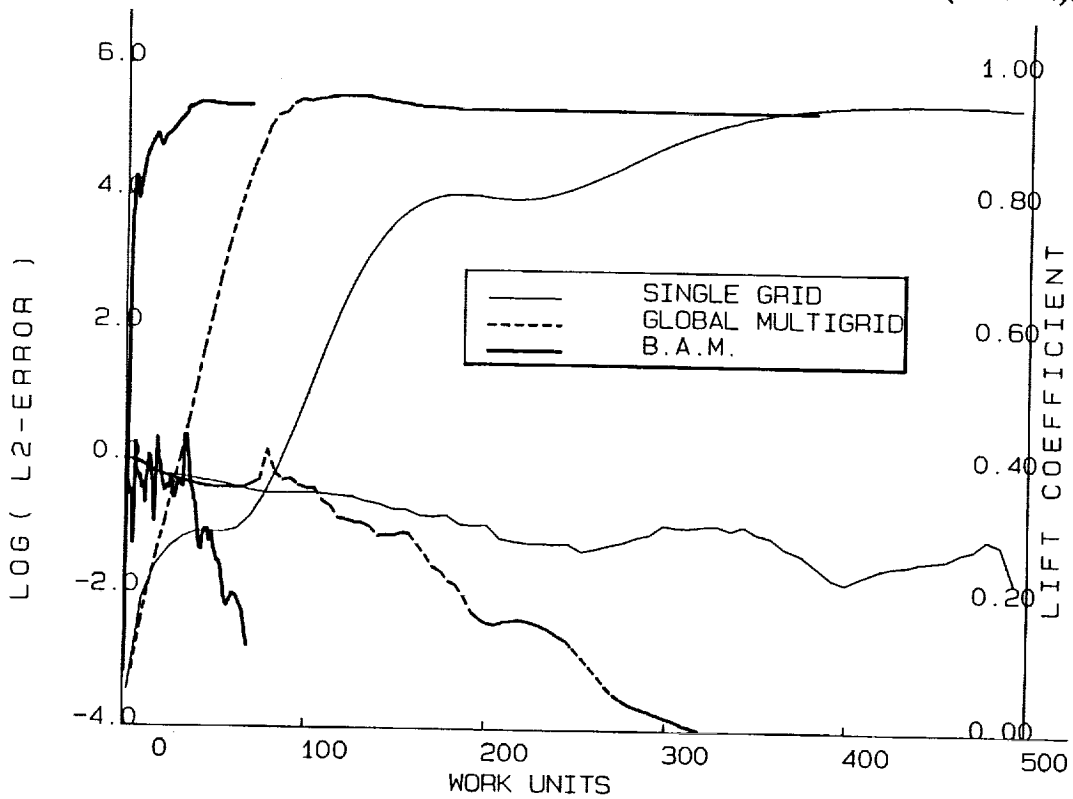

Figure 10. Convergence histories of the lift coefficient and the logarithmic Euclidean error reduction for the single grid, global multigrid and B.A.M. method (CASE 2).

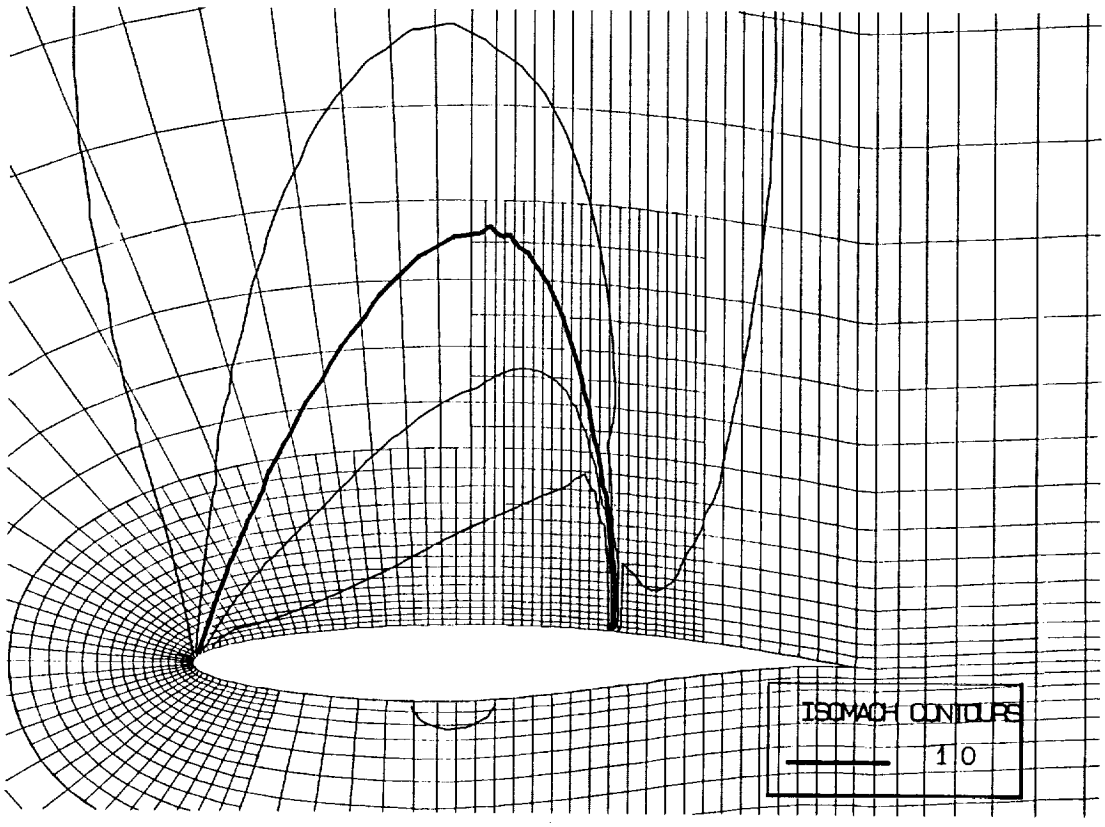

Figure 11. The adaptive composite grid and the corresponding Mach contours (CASE 2).

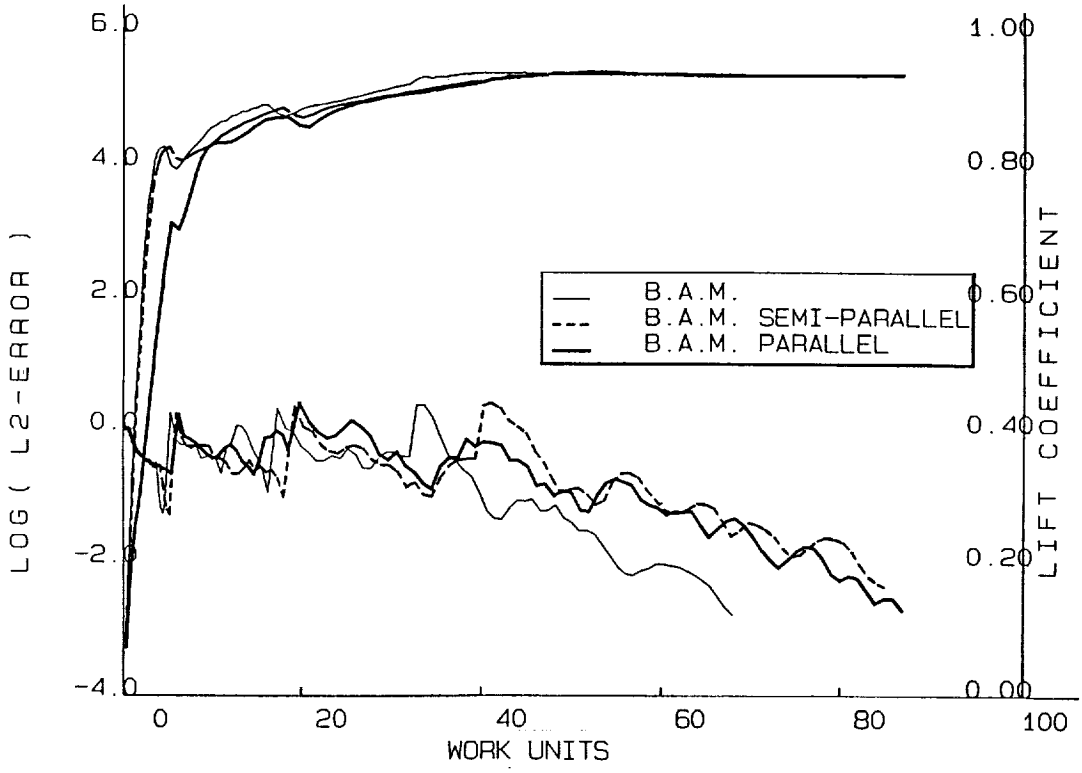

Figure 12. Convergence histories of the lift coefficient and the logarithmic Euclidean error reduction of the B.A.M. method (CASE 2) for sequential processing, "parallel" horizontal and "parallel" vertical schemes.